An Interactive Approach to Outlier Detection

Konijn, R.M.; Kowalczyk, W.

published in
Lecture Notes in Computer Science
2010

DOI (link to publisher)
10.1007/978-3-642-16248-0_54

document version
Publisher's PDF, also known as Version of record

An Interactive Approach to Outlier Detection

R.M. Konijn and W. Kowalczyk

Department of Computer Science, Vrije Universiteit Amsterdam
{rmkonijn,wojtek}@few.vu.nl

Abstract. In this paper we describe an interactive approach for finding outliers in big sets of records, such as collected by banks, insurance companies, web shops. The key idea behind our approach is the usage of an easy-to-compute and easy-to-interpret *outlier score* function. This function is used to identify a set of potential outliers. The outliers, organized in clusters, are then presented to a domain expert, together with some context information, such as characteristics of clusters and distribution of scores. Consequently, they are analyzed, labelled as *non-explainable* or *explainable*, and removed from the data. The whole process is iterated several times, until no more interesting outliers can be found.

1 Introduction

Outlier detection has many applications like fraud detection, medical analysis, etc. A successful approach to detecting anomalies must involve a deep interaction between domain experts and a powerful system that can quickly analyze huge volumes of data, present preliminary findings to experts and process their feedback. In this paper we propose such an interactive system that supports the process of finding unusual records in huge databases.

The paper is organized as follows. In the next section we provide a brief overview of the relevant research on outlier detection. Then we outline our approach to the problem, which is followed by the definition of an *outlier score function*, which is the key element of our system. This function is used to measure the "degree of deviation" of every record. Outlier scores can be calculated very quickly: it takes only two passes through the data to score all records. Next, we elaborate on the interactive aspects of the outlier detection process and provide two more mechanisms to support domain experts: a graphical representation of interesting clusters and a *local outlier score function* which refines the search process even more. Finally, we illustrate our approach on two data sets: the well-known *Abalone* set and a non-public set of 35 million of records from an insurance company.

2 Related Work

The method described in this paper can be categorised as unsupervised outlier detection, with respect to a target variable. Existing methods for unsupervised outlier detection (a missing label problem) can be split into two main categories: statistical methods and distance-based methods, which can be further split into depth-based and density-based methods. Additionally, any clustering algorithm can be used for detecting outliers.

J. Yu et al. (Eds.): RSKT 2010, LNAI 6401, pp. 379–385, 2010.

Most statistical methods use as a starting point a model of probability density function which provides a reference for measuring the degree of deviation: outliers are instances with very low probabilities. The models of pdf's might be *parametric* or *semi-parametric*, like *mixture models*, [1], [5].

The model of a pdf can be used for detecting outliers in a yet another way: the bigger the impact of an instance on the model the more unusual the instance is. This approach is used in [13], where the authors describe an algorithm called SmartSifter that uses this idea to calculate outlier scores for each instance. The outlier score of an instance is defined as the Hellinger distance between two probability distributions of available data: one built for the whole data set and the other one built for the whole data set without the observed instance. In [12], the authors create labelled examples by giving positive labels to a number of higher scored data and negative labels to a number of lower scored data. Then, with the use of supervised learning an outlier filtering rule is constructed. The data is filtered using the constructed rule, and SmartSifter is run again.

Other statistical methods include the regression and time series models. For example, when a linear regression model is fitted to data then the residual can be used to determine if an instance is an outlier or not. If there are outliers in the explanatory variables as well as the target variable, a technique called *robust regression*, [10], can be used for identifying them. There is a lot of literature about outlier diagnostics in regression models, e.g., [6] and [3].

Distance based methods require a distance measure to determine the distance between two instances. The main idea is that the distance between outlying instances and their neighbours is bigger than the distance between normal instances and their neighbours, [7]. To handle differences in densities within a dataset, the Local Outlier Factor (LOF), [4], or the Connectivity-based Outlier Factor (COF), [11], or the Multi-granularity Deviation Factor (MDEF), [8], can be used.

Another approach to outlier detection, which is described in [14], assumes that some outliers are identified before starting the search process. First, the remaining instances are ranked by a score which is based on the Multi-granularity Deviation Factor (MDEF), [8], where a high score indicates a high probability of being an outlier. Next, a training set is formed from records which have low scores (i.e., they are not likely to be outliers) and the known outliers. Finally, a Support Vector Machine model is trained on this set and applied to the remaining records to identify new outliers. The whole process is repeated several times until no new outliers can be found.

3 Our Approach

Let us consider a collection D of n records with $k+1$ fields, and let us identify one field that for some reasons is important for us. Let Y denote this selected field, and let $X_1, X_2, \ldots, X_k$ denote the remaining fields. For example, in the context of credit card transactions the selected field might represent the amount of money involved in the transaction, while the remaining fields could contain other characteristics of the transaction such as the age and the gender of the cardholder, the time and the location of the transaction, type of the terminal, etcetera. Our data set can be viewed as a sample of n observations drawn from an unknown probability distribution $P(Y, X_1, \ldots, X_k)$,

where we slightly misuse the notation using the same symbols to denote random variables and the corresponding fields of our database.

Finally, we will assume that all variables $X_1 \ldots X_k$ are discrete and take a few values; otherwise we discretize them with help of any suitable discretization method. Possible values of variable X_i will be denoted by v_{ij}, where j ranges between 1 and n_i. We will use C_{ij} to denote the set of all records for which $X_i = v_{ij}$. Sometimes we will call the sets C_{ij} *clusters*. Let us notice that for any i, clusters $C_{ij}, j = 1, \ldots, n_i$, form a partitioning of D into n_i disjoint sets.

The key idea behind our approach is based on the concept of an *outlier score function* which, for a given record $(y, x_1, \ldots, x_k)$, provides a heuristic estimate of the probability $P(Y = y|X_1 = x_1, \ldots, X_k = x_k)$, in case Y is discrete, or $P(Y > y|X_1 = x_1, \ldots, X_k = x_k)$, in case Y is continuous.

In the next section we will propose several outlier score functions that are easy to compute and have simple interpretation.

When a score function is fixed, the outlier detection process is organized in the following loop, which is repeated until no new outliers can be identified:

- Calculate outlier score for each record.
- Present to a domain expert the scored data as a collection of sets C_{ij} together with some additional statistics, so (s)he could quickly identify groups of "explainable outliers" and remove them from the data.
- Calculate for each remaining record a *local outlier score* and present to the expert all records with scores smaller than a pre-specified threshold.

In the following two sections we will describe all steps in more detail.

4 Outlier Score

Let us assume for a moment that the variable Y is discrete and let us consider a record $(y, x_1, \ldots, x_k)$ from our data set D. We would like to estimate the probability of observing y in combination with values $x_1, \ldots, x_k$, i.e.,

$$P(Y = y|X_1 = x_1, \ldots, X_k = x_k).$$

We are interested in an estimate that could be quickly calculated and conceptually simple, so it could be explained to domain experts with little or no statistical knowledge. This is achieved by estimating, for every i, $p_i = P(Y = y|X_i = x_i)$ by the ratio:

$$p_i = |(Y = y)\&(X_i = x_i)| \,/\, |X_i = x_i|.$$

where $|F|$ denotes the number of records from D that satisfy formula F, and then defining the outlier score as the product of these "one dimensional" estimates:

$$OutlierScore((y, x_1, \ldots, x_k)) = \prod_{i=1}^{k} p_i.$$

Let us notice that our score function can be computed very quickly: it takes only two scans of the data set D to score every record. The first scan is used to determine the frequencies that are needed to estimate all p_i's, and in the second scan scores of all instances are computed with help of these estimates.

In case the target value Y is continuous it is no longer possible to estimate $P(Y = y|X_i = x_i)$ by counting. However, because we are interested in extreme values of Y, we can determine an upper bound for the probability $P(Y > y|X_i = x_i)$, with help of Chebyshev's inequality, [9], which bounds, for any data sample, the ratio of observations that deviate from the sample mean by more than d sample standard deviations by $1/d^2$. In case of one-tailed estimate, this bound is $1/(1 + d^2)$.

This leads to the following definition of the outlier score function:

$$OutlierScore((y, x_1, \ldots, x_k)) = \prod_{i=1}^{k} 1/(1 + d_i^2),$$

where d_i denotes the standardized score of y with respect to all records from D that satisfy $X_i = x_i$.

More precisely, d_i is calculated in the following way. First, we find all the records in D with $X_i = x_i$ and consider the set Y_i of all y's that occur in these records. Next, we calculate the mean and the standard deviation of Y_i, $mean(Y_i)$ and $std(Y_i)$. Finally, we let: $d_i = (y - mean(Y_i))/std(Y_i)$.

Obviously, when we are interested not only in extremely large values of Y, but also in extremely small values, we should use an alternative formula for the outlier score:

$$OutlierScore((y, x_1, \ldots, x_k)) = \prod_{i=1}^{k} min(1, 1/d_i^2).$$

Values of the $OutlierScore$ function are usually very small, especially for large values of k. Therefore, we will usually work with a logarithm of this function.

5 Interactive Outlier Detection

When all records are scored a domain expert should analyze the results, identifying two groups of outliers: *explainable outliers*–outliers which are not interesting, and *unexplainable outliers*–outliers that require further investigation. In order to support the expert in the process of evaluating outliers, we will proceed in 2 steps.

Step1: Elimination of explainable outliers
First, we analyze the distribution of outlier scores and decide how to set the threshold: records with scores below the threshold are considered to be outliers.

Next, for every cluster C_{ij} we find two numbers: the cluster size (the number of records in the cluster) and the percentage of outliers in this cluster. We use these numbers as the x and y coordinates to visualize properties of all clusters in a scatter plot. We are particularly interested in "big clusters" with a "high percentage" of outliers, i.e., those that are located in the upper right corner of the plot.

Finally, the domain expert can start exploring individual clusters, starting with those that are located on the Pareto front (they are most interesting). For every selected cluster a sorted list of outliers is presented to the user, so (s)he could quickly decide whether they are interesting or not. In practice, big clusters with a high percentage of outliers are not interesting–they usually contain outliers that can be easily "explained" by the domain expert, and such clusters could be excluded from further analysis.

Step2: Identification of non-explainable outliers
At this stage the most interesting (non-explainable) outliers from those that survived Step 1 are identified. The selection is guided by a *local outlier score* which is defined as follows. For every cluster C_{ij} we calculate standardized outlier score for all the remaining outliers. In other words, we calculate the mean and the standard deviation of all scores of records that belong to the cluster (therefore the term "local") and then standardize scores of records we are interested in (outliers that survived Step 1). We call these standardized scores "local outlier scores". Finally, all outliers are sorted by the local outlier score and the top N are presented to the expert for further evaluation.

After completing both steps, i.e., the identification and removal from the data some of the explainable and non-explainable outliers, the whole process (Step 1 and Step 2) is iteratively repeated until no more interesting outliers can be found. Obviously, after each iteration the set of remaining records is changed and consequently the outlier scores have to be recalculated.

6 Results and Analysis

We implemented our method for different sets of insurance data. Because these datasets are not public, we can not show detailed results. Therefore we demonstrate the working of our algorithm with the use of the Abalone dataset that is available at the UCI data repository. After that a summary of the results obtained with the insurance data will be given.

6.1 Abalone Data

The Abalone dataset, [2], contains information about characteristics and measurements of individual abalones. The dataset consists of 4177 records with the following fields: *Sex*, *Length*, *Diameter*, *Height*, *Whole weight*, *Shucked weight*, *Viscera weight*, *Shell weight*, and *Rings*. In our experiments we used the variable *Whole weight* as target and discretized all remaining continuous variables into 20 categories with equal frequencies.

After applying Step 1 of our algorithm, it turned out that there were no big clusters of outliers we could eliminate. Next, Step 2 of our algorithm is applied. For the top m (we take $m = 50$) outliers, those with the smallest values for the global outlier score, we calculate the local outlier score. We examine the top outliers found. In the Abalone dataset, the top 3 records that have the highest local outlier score, coincide with the top 3 outliers with respect to the global outlier score. An example of one of these outliers is presented on Figure 1 which contains two scatter plots of the most relevant attributes.

6.2 Experimental Results on Health Insurance Data

To test the algorithm on a big real-life data set, we use a health insurance data set. The data contains 35 million pharmacy claims. A single record contains characteristics of the medication, patient, doctor who prescribed the medication, and the pharmacy itself. We are interested in outliers with respect to the variable *costs* (the claim amount), which will be our target variable Y.

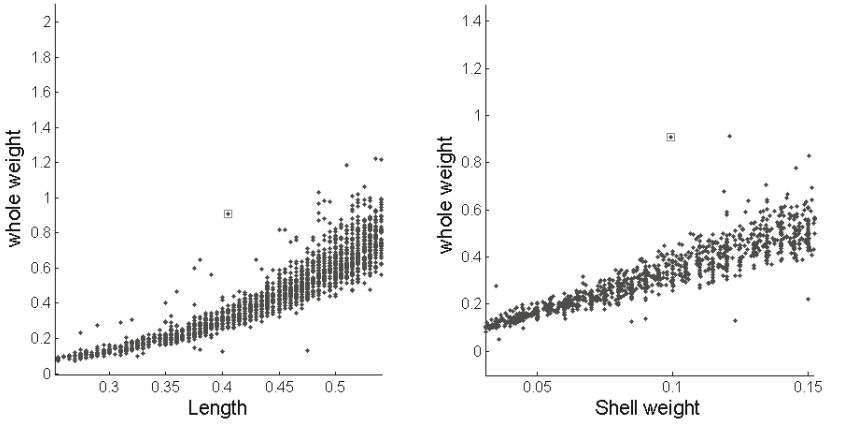

Fig. 1. Outlier 1, record number 2729 of the Abalone data set. On the left figure, the target variable *Whole weight* is plotted against *length* while the figure on the right contains the plot of *Whole weight* against *Shell weight*. The two scatter plots are zoomed-in on the outlier. Note that the scales of the y-axes on both graphs are not the same. The outlier is marked with a square around it. It can be seen that this point is an outlier.

First, we apply Step 1 of the evaluation process. For each cluster we calculate the number of outliers and the percentage of outliers. We plot them in a figure. After removing the explainable outliers, the top 50 outliers were determined with help of the local outlier score function.

In order to compare our approach to the regression-based approach, we constructed a regression tree for the data and determined some outliers by analysing standardized (per cluster) residuals. It turned out that many of the outliers found by both methods were the same. However, the proposed method was able to find small clusters or single points for which the outlying value of the target variable was "caused by" an outlying value of one predictor. For example, several records with an extremely high number of delivered pills and the corresponding extremely high costs were not detected by the regression method (the costs were proportional to the number of pills), while they were detected by our method.

7 Conclusion and Recommendations

The described method works well in practice. It is fast and therefore it can be applied in an interactive way to big data sets. It can be used while literally sitting next to a domain expert, anlyzing and removing outliers on the fly. Furthermore, it turned out to be easy to explain the method to non-statisticians and to implement it in a general purpose data analysis tools like Matlab and SAS. A possible extension of the method, especially if there is an (almost) linear relationship between an explanatory variable X and the target variable Y, is to use the two variables at the same time as target variables. Further research can focus more on feature selection or feature elimination. The method

could also be extended to operate on aggregated records instead of single records only (e.g., finding outlying patients instead of single claims). In that case feature extraction and feature selection becomes even more important. Another extension would be to provide a set of rules to the domain expert, instead of single clusters.

References

1. Agarwal, D.: Detecting anomalies in cross-classified streams: a bayesian approach. Knowl. Inf. Syst. 11(1), 29–44 (2006)
2. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
3. Belsley, D.A., Kuh, E., Welsch, R.E.: Regression Diagnostics: Identifying Influential Data and Sources of Collinearity. Wiley and sons, Chichester (2004)
4. Breunig, M.M., Kriegel, H.-P., Ng, R.T., Sander, J.: Lof: identifying density-based local outliers. SIGMOD Rec. 29(2), 93–104 (2000)
5. Eskin, E.: Anomaly detection over noisy data using learned probability distributions. In: Proceedings of the International Conference on Machine Learning, pp. 255–262. Morgan Kaufmann, San Francisco (2000)
6. Jensen, D.R., Ramirez, D.E., Jensenandd, D.R., Ramirez, E.: Bringing order to outlier diagnostics in regression models. In: Proceedings of Recent advances in outlier detection, summer research conference in statistics/ASA (2001)
7. Knorr, E.M., Ng, R.T.: Algorithms for mining distance-based outliers in large datasets. In: VLDB 1998: Proceedings of the 24rd International Conference on Very Large Data Bases, pp. 392–403. Morgan Kaufmann Publishers Inc., San Francisco (1998)
8. Papadimitriou, S., Kitagawa, H., Gibbons, P.B., Faloutsos, C.: Loci: Fast outlier detection using the local correlation integral. In: International Conference on Data Engineering, p. 315. IEEE Computer Society, Los Alamitos (2003)
9. Ross, S.M.: Introduction to Probability and Statistics for Engineers and Scientists, 4th edn. Elsevier Academic Press, Amsterdam (2009)
10. Rousseeuw, P., Leroy, A.: Robust regression and outlier detection. Wiley and Sons, Chichester (2003)
11. Tang, J., Chen, Z., chee Fu, A.W., Cheung, D.: A robust outlier detection scheme for large data sets. In: Proceedings 6th Pacific-Asia Conf. on Knowledge Discovery and Data Mining, pp. 6–8 (2001)
12. Yamanishi, K., Takeuchi, J.-i.: Discovering outlier filtering rules from unlabeled data: combining a supervised learner with an unsupervised learner. In: KDD 2001: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 389–394. ACM, New York (2001)
13. Yamanishi, K., Takeuchi, J.-I., Williams, G., Milne, P.: On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms, pp. 320–324 (2000)
14. Zhu, K., Papadimitriou, F.: Example-based outlier detection with relevance feedback. DBSJ Letters 3(2) (2004)